ACEEE Int. J. on Communications, Vol. 4, No. 1, July 2013

Improving Performance of TCP in Wireless Environment using TCP-P

Ranjeet Vasant Bidwe[1] and Amar Buchade[2]
Pune Institute of Computer Technology,Pune.
ranjeetbidwe@hotmail.com[1],amar.buchade@gmail.com[2]

***Abstract*- Improving the performance of the transmission control protocol (TCP) in wireless environment has been an active research area. Main reason behind performance degradation of TCP is not having ability to detect actual reason of packet losses in wireless environment. In this paper, we are providing a simulation results for TCP-P (TCP-Performance). TCP-P is intelligent protocol in wireless environment which is able to distinguish actual reasons for packet losses and applies an appropriate solution to packet loss.**

TCP-P deals with main three issues, Congestion in network, Disconnection in network and random packet losses. TCP-P consists of Congestion avoidance algorithm and Disconnection detection algorithm with some changes in TCP header part. If congestion is occurring in network then congestion avoidance algorithm is applied. In congestion avoidance algorithm, TCP-P calculates number of sending packets and receiving acknowledgements and accordingly set a sending buffer value, so that it can prevent system from happening congestion. In disconnection detection algorithm, TCP-P senses medium continuously to detect a happening disconnection in network. TCP-P modifies header of TCP packet so that loss packet can itself notify sender that it is lost.

This paper describes the design of TCP-P, and presents results from experiments using the NS-2 network simulator. Results from simulations show that TCP-P is 4% more efficient than TCP-Tahoe, 5% more efficient than TCP-Vegas, 7% more efficient than TCP-Sack and equally efficient in performance as of TCP-Reno and TCP-New Reno. But we can say TCP-P is more efficient than TCP-Reno and TCP-New Reno since it is able to solve more issues of TCP in wireless environment.

***Index Terms*—TCP, Wireless networks, Protocol Design.**

I. Introduction

Communication networks are experiencing a great value now days. Packet switching technologies have eventually merged the traditional voice networks and data networks together. So they can able to form a converged and integrated multimedia network which is further extended to incorporate with wired, wireless and satellite technologies. All-Internet protocol (IP [1]) wired and wireless hybrid network is becoming a reality. In this reality, we can say Transmission control protocol (TCP [2]) is a dominant communication protocol, as it is carrying about 90% internet traffic.

TCP was designed originally for wired networks. But now, TCP needs to depart from its original wired network oriented design and must try to evolve to meet the challenges in wireless environments. Network layer protocol (IP) is best-effort-based variable length packet delivery and connectionless protocol. But network layer protocol does not guarantee the timely and in-order delivery of packets between end stations and also not guarantee reliable data transfer.

TCP is a transport layer protocol that uses the basic IP services to provide applications with an end-to-end connection- oriented packet transport mechanism that ensures the reliable and ordered delivery of data. TCP implement flow control and congestion control algorithms [3] based on the sliding window and additive increase multiplicative decrease (AIMD) algorithms.

TCP is having sliding window-based flow control mechanism. This mechanism allows the sender to advance the transmission window upon the receptionof an acknowledgment (ACK) that indicates the last packet has been received successfully by the receiver. When packet is lost because of any reason, either the sender receives duplicate ACKs (DUPACK) from receiver or the sender's retransmission timeout (RTO) timer expires, which results in retransmission from sender side. Such events activate the sender's congestion control mechanism. In this mechanism sender reduces the size of its transmission window, or congestion window, which is called as *cwin* in TCP term, resulting in a lower transmission rate to relieve the link congestion.

The basis of TCP congestion control lies in the following algorithms: slow start, congestion avoidance, fast retransmit and fast recovery. In TCP, the error recovery strategies employ the congestion control scheme in the following way. TCP continually measures how long the acknowledgments take to return. It maintains a average of this delay (Round Trip Time-RTT) and an estimate of the expected deviation in delay from the average (Delay Variation). If the current delay is longer than the average by more than four times the expected deviation, TCP employs congestion avoidance as a way to deal with lost packets.

II. Related Work

Results of TCP in wired environment are pretty good because in wired environment reason for packet loss is congestion only. But this thing is not true in wireless environment, since we have number of reasons in wireless environment for packet losses [3]. Main problem of TCP is it is not able to identify original reason for packet losses. TCP replies each

DOI: 01.IJCOM.4.1.1256

packet loss by congestion control algorithms, which results in degradation of performance of TCP in wireless environment. TCP is facing different issues like Congestion [4] in network, Disconnection in connections, Available Bandwidth [5] in wireless environments, Mobility and Handoff [6], Random packet losses [7], High Bit error rates, big Round Trip Time(RTT) etc in wireless environment. Below we are giving brief about TCP versions which try to solve mentioned issues.

Freeze TCP [8] is try to identify happening disconnection in links due to potential handoff, fading signal strength or any other reason due to wireless media. Here receiver is taking care of it. Receiver is sensing link continuously and identifies happening disconnection in advance. If signal is fading then it notifies sender by acknowledgement message so that sender stop sending message to receiver. When connection is established again then receiver again notifies sender. While notifying sender, receiver sends acknowledgement for last received packet so that sender starts sending next packets. During disconnected period sender checks about connection from receiver by probe messages. But main dis-advantage of Freeze-TCP is that, Freeze-TCP is only useful, if a disconnection occurs while the data is being transferred. It is not useful, in case of a disconnection when no data is being transferred between sender and receiver.

The original TCP i.e. TCP-Tahoe [9] consisting of three transmission phases, namely slow-start, congestion avoidance, and fast retransmit. TCP-Reno [10] is new version adds fast recovery phase to TCP-Tahoe. When TCP is starts sending packets, it maintains two variables, the cwnd(congestion window size), which is initially set to be 1 maximum segment size (MSS), and ssthresh (SS threshold). In beginning of connection, source enters in slow-start phase. In this phase source increases cwnd value by 1 MSS after receiving each ACK, means sender's cwnd value grows exponentially. When cwnd value equals ssthresh value, sender enters in congestion avoidance phase. In this phase cwnd value is increased by one on each ACK received. This additive increase leads to the linear growth of the transmission rate that helps the sender to slowly probe the available network bandwidth. When sender receives DUPACK's (generally number is 3), then congestion window is reduced by half of value. Here TCP assumes that packet loss is because of congestion. Then sender sets ssthresh value same as cwnd value, and starts retransmission of lost packets. This is beginning of fast retransmit phase. TCP-Reno invokes the fast recovery algorithm to speed up the recovery process, by which the sender treats the DUPACKs received during the fast retransmit phase as normal ACKs and artificially inflates the cwnd value. This inflated portion of cwnd is later deducted at the end of the fast retransmit phase. When sender receives normal ACK, then fast retransmit phase is ended. This last ACK acknowledges sender that receiver has successfully received an ordered packet whose sequence number passes. Therefore, TCP decreases its cwnd value multiplicatively in the presence of packet loss.

Modified version of TCP-Reno is TCP-New Reno [11]. TCP-New Reno much more efficient that RENO. New Reno is able to detect multiple packet losses. TCP-New Reno works similar as TCP-Reno in fast transmission phase, changes are made just in fast recovery phase. When New-Reno enters fast recovery phase, it first checks the maximums segment which is outstanding. However when fresh ACK is received then two cases are considered, first if received ACK is acknowledging all outstanding segments then fast recovery phase is ended and again congestion avoidance phase is started. Second If received ACK is partial, then next packet is sent and number of duplicate acknowledgments is set to zero. It exits fast recovery when all the data is acknowledged. Problem regarding New-Reno is it requires one RTT to detect each packet loss happened.

Next proposed version, TCP Westwood [12] is a sender-side modification of the TCP congestion window algorithm. It improves upon the performance of TCP Reno in wired as well as wireless networks. In TCP Westwood bandwidth estimate (BWE) to set the cwnd and ssthresh after congestion episode. TCP Westwood is different from TCP Reno because, TCP Reno halves the congestion window after three acknowledgments where TCP Westwood attempts to select asssthresh and cwnd which are consistent with the effective bandwidth used at the time congestion is experience. The source performs end-to-end estimate of the bandwidth available along a connection between sender and receiver. Sender uses the estimated value of bandwidth to properly set the cwnd and ssthresh. This way TCP Westwood avoids overly conservative reduction of cwnd and ssthresh; and thus it ensures faster recovery.

Better throughput, good put and delay performance, fairness as well as friendliness when coexisting with TCP Reno were observed in experimental studies of TCP Westwood. TCP Westwood has only one disadvantage that it performs poorly when random packet loss rate exceeds a few percent. TCP-Westwood+ [13] is next improved of TCP Westwood, but it was not successful since it was not working well in the presence of reverse traffic due to ACK compression.

TCP-Jersey [14] is an end-to-end approach modification, it adds two improvements, the available bandwidth estimation (ABE) algorithm and the congestion warning (CW) router configuration. ABE is tried to calculate more accurate time varying network bandwidth. Here sender monitor the rate of receiving ACKs, and then sets the optimum congestion window size based on this calculations. When congestion is happening, sender adjusts its transmission rate.

TCP-Jersey computes the optimum congestion window once every RTT using a modified sliding window algorithm. CW is congestion warning sent to sender configuration, so that it can adjust packet sending rate according to network congestion. The purpose of CW is to convey a alert of the bottlenecked queue to the sender. Also, the marking of packets by the CW configured routers helps the sender of the TCP connection to effectively differentiate packet losses caused by network congestion from those caused by wireless

DOI: 01.IJCOM.4.1.1256

link errors.

III. TCP-P Approach

TCP-P [15] stands for TCP-Performance. Talking about basic TCP functionalities, then yes, TCP-P satisfies true End-to-End semantics of TCP since no intermediaries involved. It provides reliable, connection oriented service for mobile nodes. TCP-P uses the standard TCP mechanisms for flow control and connection management. Mainly TCP-P tries to solve threeimportant issues of TCP that are Congestion,.So, TCP-P Disconnection and Random Packet Losses mainly having three functionalities. Working with these three functionalities TCP-P is able to detect packet losses due to congestion in network, disconnection in network links and random lost packets. TCP-P is more successful than other TCP versions since it is having more Packet Delivery Ratio as well as able to solve more issues. We will see those result values in next section. Now we see each functionality of protocol in brief. Figure 1 provides diagrammatic view about TCP-P working.

A. Congestion Avoidance Algorithm

It is a sender side modification algorithm. The main advantage of using this algorithm is that, it is able to detect happening congestion without a packet loss. When data sending is going on at the same time, sender is continuously computing the connection Bandwidth Estimate (BWE) which is equal to the rate at which data is delivered to the TCP receiver. The BWE value is computed by performing end-to-end estimate of the bandwidth available along with the TCP connection by measuring and averaging the rate of returning ACKs. This estimated BWE value is used to set congestion window (cwnd) and slow start threshold (ssthresh) before congestion episode. Whenever sender perceives a packet loss (i.e. a timeout occurs or 3 duplicate ACKs are received), the sender uses the BWE to properly set the congestion window (cwnd) and the slow start threshold (ssthresh) and sends data accordingly. This mechanism very little bit different from slow start mechanism. In slow start mechanism, packet sending rate is increased exponentially i.e. if 2 packets delivered successfully, then it will try for 4 packets, then try for 8 and so on. But in this approach packet sending rate is not increased exponentially i.e. if 2 packets delivered successfully, but it is not able to delivered 4 packets, can only delivered 3 packets then it will try for sending 3 packets only, not 4 packets. It will prevent system from loss of packets.

B. Disconnection Detection Algorithm

This time receiver is working to detect the things. Here receiver senses wireless medium continuously for detecting fading signals which in turn detects happening disconnection. In certain cases, it might even be able to predict a temporary disconnection (signal strength is fading for instance). In such a case, it advertises a zero window size, then it forces sender into the ZWP mode and prevent it from dropping its congestion window. When the receiver senses an impend ing disconnection, first it advertises its window size as zero and a zero window acknowledgement (called as ZWA)to sender prior to disconnection to inform sender about disconnection. This period is called as "warning period" pro

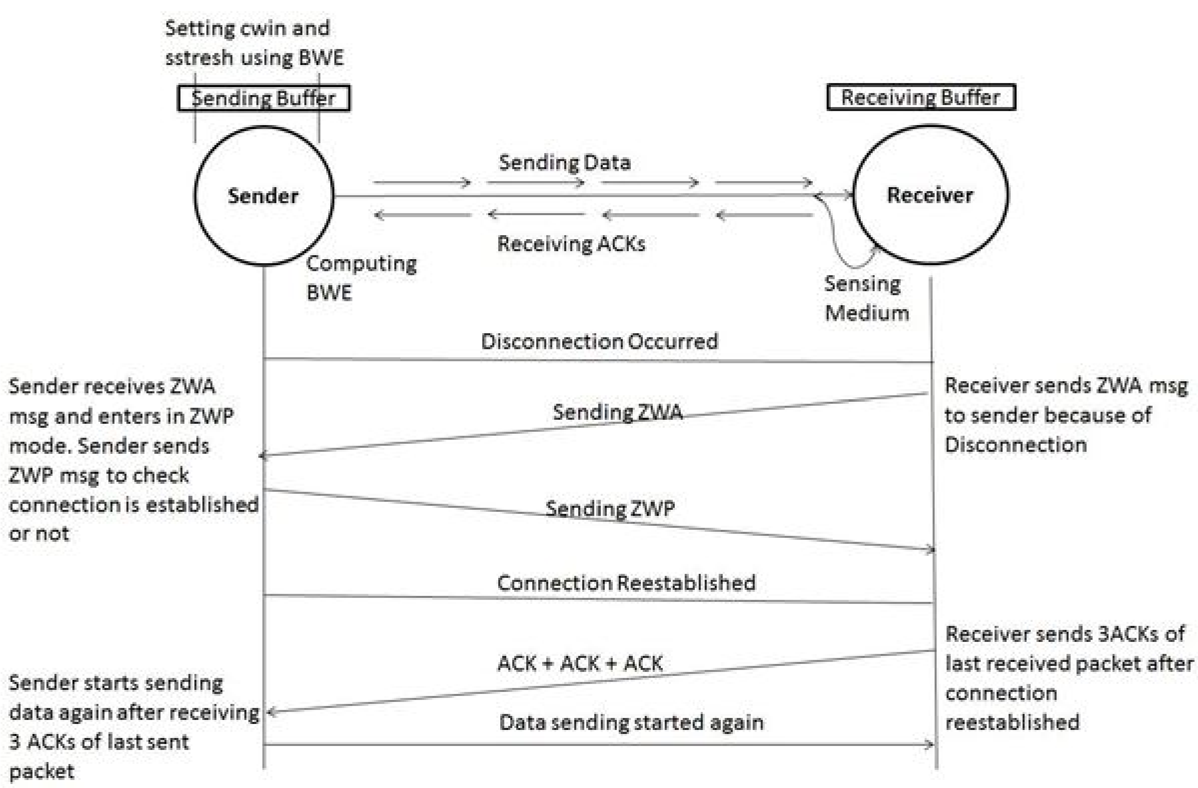

Figure 1. TCP-P Approach [15]

DOI: 01.IJCOM.4.1.1256

vided that warning period should be long enough than time required for one ZWA to get across sender).If warning period is any longer, then sender is forced into ZWP mode. If warning period is small then receiver will not have enough time to inform sender, and sender have to drop its congestion window. When connection is established again then receiver sends three ACKs for last received packet and sender starts sending data again. To check connection is established or not sender sends zero window probes to receiver after an interval of time.

C. Randomly Lost Packets Detection

Wireless links are highly unreliable and they lose segments all the time due to a number of factors. According to [16], noise in network is main reason behind randomly lost packets. Up to 30% of messages can be lost because of random packet losses.For randomly lost packets TCP-P also provides the solution. TCP-P just modifies the header part of the packet. When packet is lost, i.e. its lifetime exceeds TTL value of packet, and then lost packet itself sends loss notification message to sender. To gain this functionality we can modify TTL field in TCP header to send a ICMP message to sender. This message can use value from sender IP address from header part of TCP to send ICMP message. By this way sender can detect lost packet and resends same packet again to receiver.

IV. Results And Observations

This section is providing performance results of TCP-P compared to other successful TCP versions in wireless environments. We have used Network Simulator (NS-2.34) on fedora 13 system to collect results. We are taking values for different number of nodes (starting from 100 to 200). We have taken all these results for simulation time of 200 ms. Here performance is calculated on basis of number of technical stuffs.

A. Packet Delivery Ratio

PDR for a protocol can be calculated by formula:

$$\mathrm{PDR} = \frac{\text{Total Number of Packets Delivered to Receiver}}{\text{Total Number of Packets Sent by Sender}}$$

Here, we have calculated PDR for each version of TCP for different Number of Nodes. Table 1 give detailed idea about analysis and Following graph is showing performance of TCP-P compared to other TCP versions with respect to its packet delivery ratio. Fig 2 is showing similar values in table. Above table shows that, TCP-P is more efficient than other TCP versions in wireless environment. Mathematically, we can say, TCP-P is 3.5% more efficient than TCP-Tahoe, 3% more efficient than TCP-Vegas, 3% more efficient than TCP-Sack and 3.5% efficient in performance as of TCP-Reno and TCP-New Reno. But we can say TCP-P is more efficient than TCP-Reno and TCP-New Reno since it is able to solve more issues of TCP in wireless environment. Another thing we can observe that, TCP-P works exceptionally well If number of nodes are equal to or greater than 100. If nodes value exceeds 100, then TCP-P is most efficient protocol to serve the system. In this case, we can say, TCP-P is more efficient than TCP-Reno and TCP-New Reno also.

A. Control Overhead

Control Overhead is a showing value of number of control messages sent in packet sending from sender to receiver. Following graph shows number of control messages sent in a simulation time of 200 ms. Control messages make difficulties to packet sending, so less number control messages will tends to more number of packets to deliver to receiver. From Table II and Fig. 3, we can say that, TCP-P is 2% more efficient than

Table I. No of Nodes Vs. Pdr

Number of Nodes	PDR (Packet delivery Ratio) for TCP.					
	TCP-P	TCP-New Reno	TCP-Reno	TCP-Vegas	TCP-Tahoe	TCP-Sack
100	99-5869	98.4134	98.6137	98.3122	97.6133	98.7941
110	86.0349	84.0372	84.037	71.6667	90.7859	90.7859
120	98.5599	98.5403	98.5476	96.5726	97.8172	98.8403
130	96.4965	96.475	96.4645	98.2603	98.3138	98.2952
140	72.4138	72.4138	72.4032	71.1112	69.2872	70.3972
150	98.5114	98.5114	98.516	99.5691	71.5122	67.8571
160	99.9345	99.9345	99.1173	99.9868	98.9872	99.8422
170	99.8451	98.8237	98.3456	97.9921	97.7176	96.5213
180	99.0153	98.8393	97.6134	98.8112	96.9281	99.172
190	99.5011	98.1113	97.1276	98.0003	94.6271	97.9156
200	99.7026	70.4918	69.8762	97.1193	95.8183	99.7022

DOI: 01.IJCOM.4.1.1256

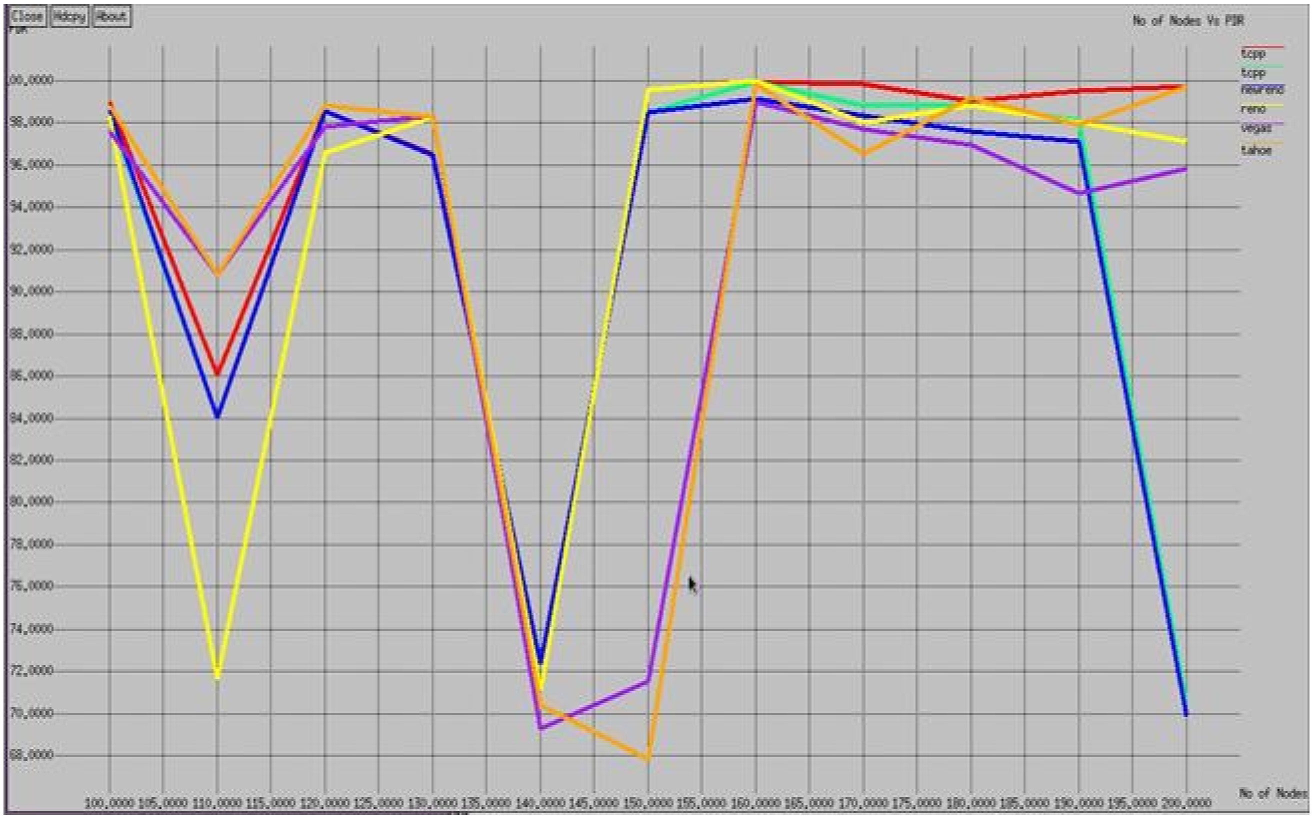

Figure 2. No of Nodes Vs. Packet Delivery Ratio

TABLE II. NO OF NODES VS. CONTROL OVERHEAD

Number of Nodes	Control Overhead for TCP					
	TCP-P	TCP-NewReno	TCP-Reno	TCP-Vegas	TCP-Tahoe	TCP-Sack
100	27854	27840	27856	6473	29762	22249
110	22413	23110	23189	29228	24767	22343
120	25578	25760	25798	25980	26776	25448
130	27718	27815	27654	27697	28762	22024
140	28119	28119	28934	29731	29119	19725
150	31620	31620	31721	29361	32633	3156
160	35914	35914	35678	39619	36972	35269
170	35817	35950	35478	36417	36118	37813
180	38643	38950	37990	39113	19690	38048
190	36160	39750	39110	39870	39980	40170
200	34832	36635	36278	39898	45672	44832

NewReno and Reno. 3% efficient than vegas, 2% efficient than Tahoe and 9% efficient than Sack.

C. Routing Overhead

Routing overhead is number of messages required for routing purpose between different nodes. It makes TCP difficult to deliver more number of packets, similar to control messages. So lesser the routing overhead, more number of packets are delivering to destination. Table III and Fig. 4 shows routing overhead for different TCP versions.While thinking of routing messages, TCP-P is more efficient. TCP-P is 1% more efficient than Newreno, 1.5% more efficient than Reno and much more efficient than other versions.

DOI: 01.IJCOM.4.1.1256

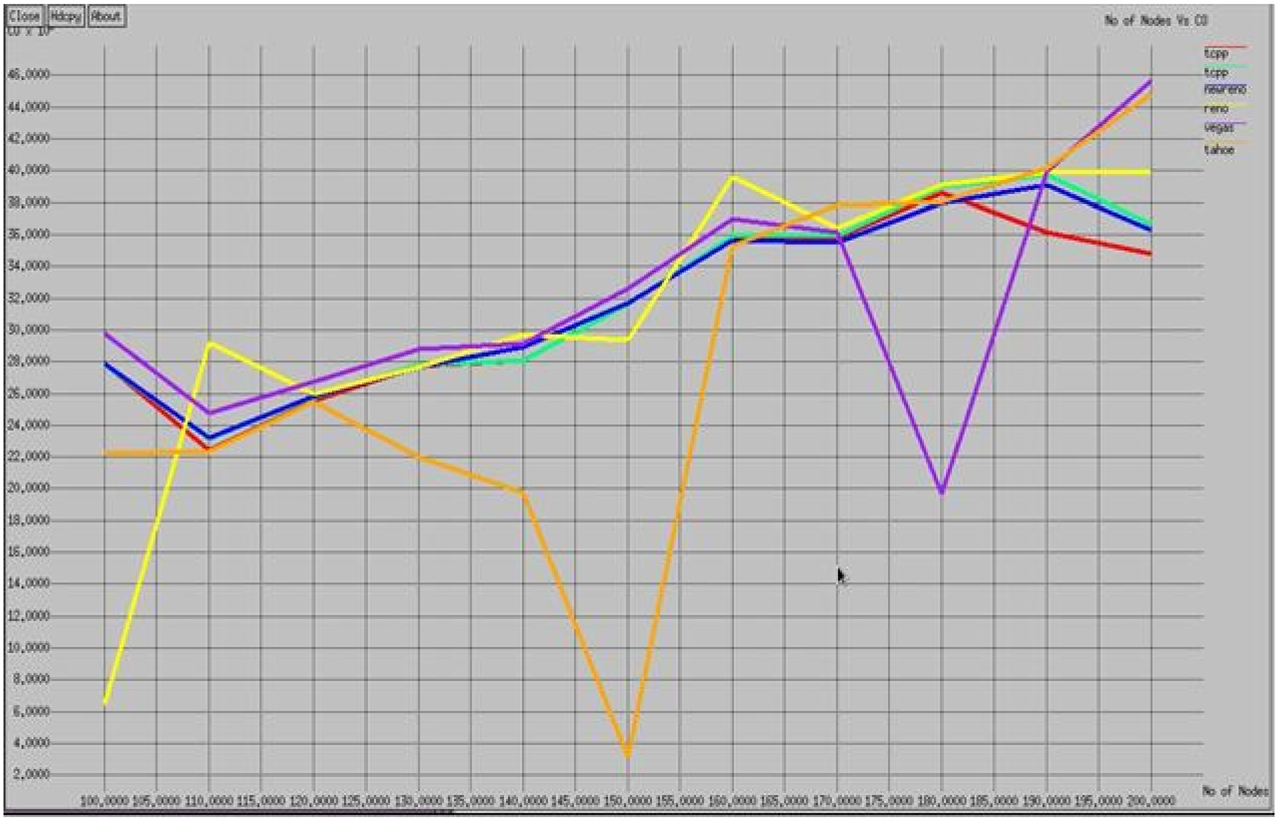

Figure 3. No of Nodes Vs. Control Overhead

TABLE III. NO OF NODES VS. ROUTING OVERHEAD

Number of Nodes	Routing Overhead for TCP					
	TCP-P	TCP-NewReno	TCP-Reno	TCP-Vegas	TCP-Tahoe	TCP-Sack
100	2.5679	2.568	2.578	5.55622	5.7621	3.87951
110	64.9652	62.9596	62.9176	679.721	831.12	66.6955
120	7.11547	9.17629	9.1834	10.8117	10.5421	9.6321
130	8.7061	10.8015	10.8432	9.80772	11.1214	8.12694
140	669.5	669.5	669.5	711.6	811.72	165.726
150	10.6179	10.6179	10.8154	11.5504	11.5511	166.105
160	4.70448	4.70448	4.71887	2.61788	6.7011	5.57525
170	6.1622	6.2014	6.2325	6.3241	7.1121	8.1273
180	8.17668	7.14679	7.15762	9.1214	8.2976	10.2472
190	5.2112	5.6773	5.6987	6.0134	7.1121	6.7112
200	4.77648	8.51997	8.52	7.2118	5.78624	4.78208

D. Delay

In simple words, we can say Delay is the time between when a packet is sent from sender and when it is received at receiver. So we must have a less delay in packet sending for an efficient system. Above graph is providing delay values for TCP-P compared to other TCP versions. From table IV and Fig. 5, we can say that TCP-P have very less delay compare to other TCP versions, means TCP-P is very significant about packet delivery.

DOI: 01.IJCOM.4.1.1256

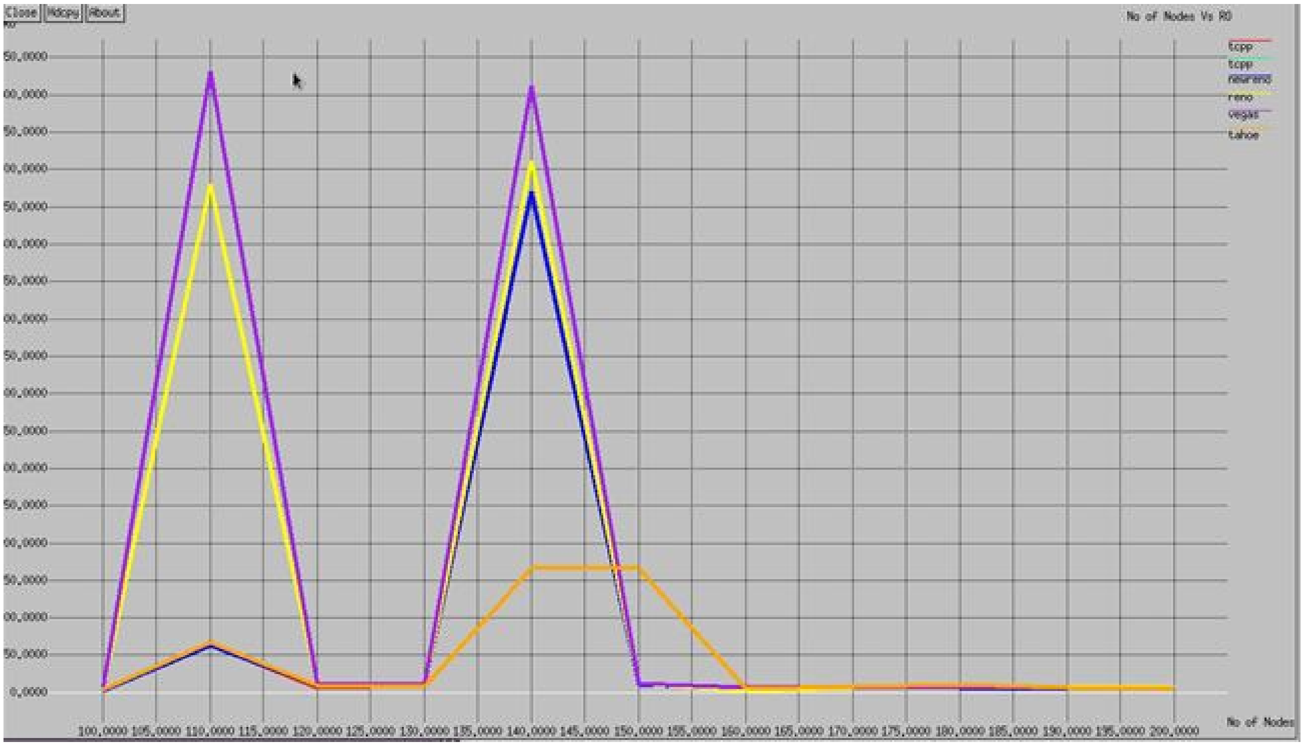

Figure 4. No of Nodes Vs. Routing Overhead

TABLE IV. NO OF NODES VS.DELAY

Number of Nodes	Delay					
	TCP-P	TCP-NewReno	TCP-Reno	TCP-Vegas	TCP-Tahoe	TCP-Sack
100	0.267163	0.271221	0.271229	0.228806	0.287264	0.265892
110	0.440942	0.450623	0.450712	2.15405	0.45767	0.519411
120	0.282946	0.282946	0.28211	0.298156	0.291137	0.267385
130	0.316443	0.316511	0.316987	0.154192	0.326541	0.290801
140	0.183756	0.183756	0.183987	0.195117	0.193756	0.523557
150	0.257201	0.257201	0.257423	0.031847	0.267116	1.03482
160	0.223811	0.223811	0.223976	0.040119	0.231138	0.265967
170	0.241156	0.242251	0.242543	0.255137	0.241156	0.261213
180	0.285542	0.257491	0.257497	0.29716	0.289167	0.23234
190	0.261522	0.268313	0.268319	0.281176	0.29118	0.279113
200	0.243738	2.15405	2.15498	0.323536	0.27672	0.24122

E. Jitter

Generally we can define jitter as the variation in the time between packets arriving at receiver, caused by network congestion, timing drift, or route changes. Lesser the jitter value indicates the efficient packet delivery at receiver. Above graph is giving values of TCP-P and other versions. Results from table V and Fig. 6, shows that TCP-P have very much less jtter value than any other TCP version.

V. CONCLUSION

In this paper, we can examine the issues related to TCP communications over wireless environments. In particular, issues like congestion, frequent disconnections, Bandwidth estimation and random packet losses. Here we improve TCP performance by proposing new intelligent solution called TCP-P. TCP-P is an intelligent protocol which is able to detect

DOI: 01.IJCOM.4.1.1256

ACEEE Int. J. on Communications, Vol. 4, No. 1, July 2013

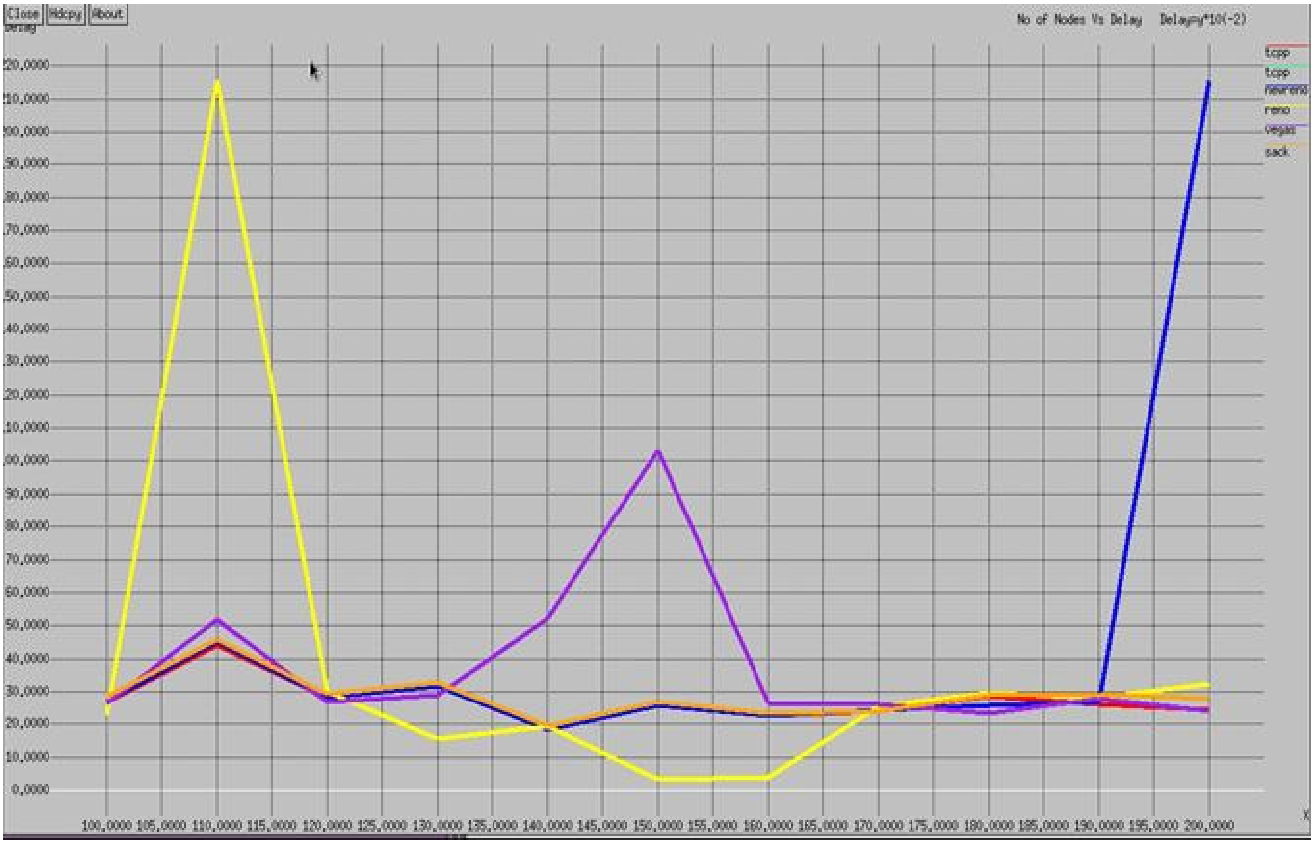

Figure 5. No of Nodes Vs. Delay

TABLE V. NO OF NODES VS.JITTER

Number of Nodes	Jitter					
	TCP-P	TCP-NewReno	TCP-Reno	TCP-Vegas	TCP-Tahoe	TCP-Sack
100	0.016626	0.017126	0.017127	0.011845	0.019118	0.022569
110	0.404983	0.410132	0.410178	3.21598	3.182176	0.091908
120	0.0516	0.516002	0.516987	0.691213	0.691213	0.019732
130	0.034644	0.035651	0.03623	0.047919	0.037614	0.018192
140	0.241938	0.241938	0.24234	0.321761	0.291872	0.26691
150	0.060474	0.060474	0.06123	0.010623	0.070431	0.293934
160	0.011435	0.011435	0.01235	0.011691	0.018446	0.137996
170	0.018122	0.018172	0.018897	0.021331	0.019133	0.019117
180	0.02776	0.032205	0.03456	0.035214	0.027862	0.017454
190	0.024427	0.247162	0.25876	0.051673	0.026121	0.026113
200	0.015355	0.03216	0.032983	0.019378	0.019314	0.015318

actual reason of packet loss, i.e. congestion, disconnection, of random packet lost. TCP-P satisfies true end-to-end semantics as no intermediaries are not involved between sender and receiver; it just requires modification in the TCP code. TCP-P controls rate of sending data by examining happening disconnection by sensing medium continuously. It is a congestion avoidance algorithm. It calculates receiving rate of ACKs from receiver and efficiently uses a bandwidth available in the medium. Also it computes bandwidth estimate value to avoid occurring congestion. By this way, it detects

DOI: 01.IJCOM.4.1.1256

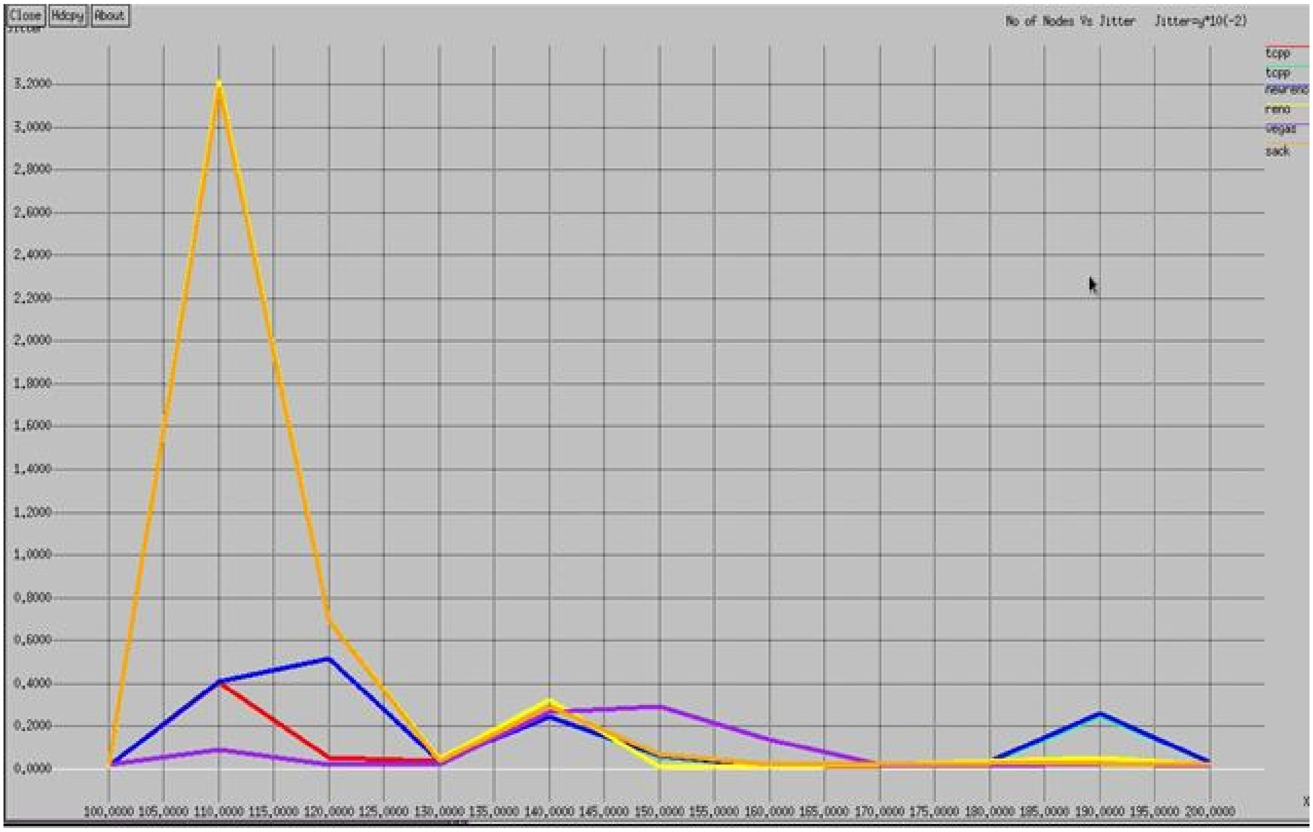

Figure 6. No of Nodes Vs. Jitter

happening congestion first and tries to avoid slow start mechanism. It modifies header part of packet, which informs sender by sending a loss notification message if packet is lost.

As per our simulation results, we can use TCP-P instead of TCP-Reno or TCP-New Reno. We can say, TCP-P is 3.5% more efficient than TCP-Tahoe, 2% more efficient than TCP-Vegas, 3% more efficient than TCP-Sack and 3.5% efficient in performance as of TCP-Reno and TCP-New Reno. Although this system gives little bit similar results as TCP-New Reno, but it is able to solve more issues of TCP in wireless environment. Also we can conclude that, if network is containing more than 100 nodes, then TCP-P is always the efficient transport protocol for particular network.

References

[1] J. Postel, “Internet Protocol,” IETF, RFC 791, 1981.

[2] “Transmission Control Protocol,” IETF, RFC 793, 1981.

[3] Ranjeet V. Bidwe, “Different Issues and Survey of Proposed Solutions in TCP over Wireless Environments.”, International Journal of Future Computer and Communication, 2012.

[4] V, Jacobson, “congestion avoidance & control”, Computer Communication Review dl.acm.org, 1995.

[5] Y Tian, K Xu, N Ansari , “TCP in wireless environments: problems and solutions”, Communications Magazine, IEEE, 2005.

[6] VassilisTsaoussidis Ibrahim Matta, “Open Issues on TCP for Mobile Computing”, Mobile Computing, Wiley Online Library, 2002.

[7] Ka- cheongleung and Victor O. K. LI”Transmission Control Protocol (TCP) in wirelessnetworks: Issues, Approaches and Challenges”, Communications Surveys &Tutorials,IEEE, 2006.

[8] Tom goff, James Moronski, D. S. Phatak, Vipul Gupta, “Freeze-TCP: A true end to endTCP enhancement mechanisms for mobile environment”, Nineteenth Annual JointConference of the IEEE Computer and Communications Societies, IEEE, 2000.

[9] W. Stevens, “TCP slow start, congestion avoidance, fast retransmit andfast recovery algorithms,” IETF, RFC 2001, 1997.

[10] C Bai, J Liu, H Deng ,”Research on Optimization Algorithm of TCP Reno Based onWireless Network”, Springer, 2011.

[11] HanaaTorkey, GamalAttiya, Ibrahim Z. Morsi, “Modified Fast Recovery Algorithm for Performance Enhancement of TCP-NewReno”, International Journal of Computer Applications, Volume 40 - Number 12, 2012.

[12]SeverioMoscola, Claudio casetti, Mario Gerla, M.Y.Sanadidi&Ren Wang, “TCPWESTWOOD: Bandwidth estimation for enhanced transport over wireless links”.international conference on Mobile computing and networking,ACM, 2001.

[13] D Kliazovich, F Granelli, “TCP Westwood+ enhancement in high-speed long-distance networks”, ICC’06. IEEE, 2006.

[14] K Xu, Y Tian, “TCP-Jersey for wireless IP communications”, Selected Areas inCommunications, IEEE, 2004.

[15] Ranjeet V. Bidwe, Amar R. Buchade, “TCP-P: A new approach in wireless environment to solve issues currently TCP facing” Second International Conference on Computational Intelligence and Information Technology – CIIT 2012, Springer, 2012.

DOI: 01.IJCOM.4.1.1256